

Research Article

Optimization of Quantitative Financial Data Analysis System Based on Deep Learning

Meiyi Liang 

Department of Finance, Dongguan University of Technology, Dongguan 523001, China

Correspondence should be addressed to Meiyi Liang; liangmy@dgut.edu.cn

Received 6 January 2021; Revised 8 March 2021; Accepted 19 March 2021; Published 28 March 2021

Academic Editor: Wei Wang

Copyright © 2021 Meiyi Liang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to better assist investors in the evaluation and decision-making of financial data, this paper puts forward the need to build a reliable and effective financial data prediction model and, on the basis of financial data analysis, integrates deep learning algorithm to analyze financial data and completes the financial data analysis system based on deep learning. This paper introduces the implementation details of the key modules of the platform in detail. The user interaction module obtains and displays the retrieval results through data parsing, calling the background, and computing engine. The data cleaning module fills, optimizes, and normalizes the data through business experience; the calculation engine module uses the algorithm and extracts the database information to get the similar time series and matching financial model. Finally, the data acquisition module fills the database with historical data at the initialization stage and updates the database every day. The data analysis platform for quantitative trading designed and implemented in this paper has carried out demand analysis, design, implementation, and test. From the perspective of function test and performance test, two functions of similar stock search and financial matching model are selected and tested, and the results are in line with the expected results.

1. Introduction

The financial market mainly includes stocks, futures, and bonds. In order to better predict its volatility, trend, and price, many experts and scholars explore the financial field by building and designing a variety of deep learning models to obtain ideal results. Stock data has always been an important group of financial data [1, 2]. The prediction of stock data is widely concerned in the field of finance, machine learning, and deep learning. Financial data is a complex, high-dimensional, and noisy time series data. The main reason is the lack of accuracy in the analysis of market psychology and financial policy [3]. For the prediction research of financial market, the accuracy of traditional financial econometric analysis methods is low, which is not suitable for complex and high-dimensional financial data. Deep neural network can simulate the form of human brain, learn and analyze the characteristics of complex data layer by layer and take it as the data form, and ultimately improve the effect of data classification and prediction [4, 5].

In different academic fields, scholars have proposed a variety of financial analysis and forecasting methods, including (1) financial forecasting based on economics and its improved forecasting methods, such as the financial forecasting method based on grey linear regression combination model, which has the characteristics of less data demand and accurate forecasting model; (2) forecasting methods and models based on mathematics and statistics, such as improved hidden Markov model and its application in financial forecasting, financial forecasting based on Bayesian maximum likelihood estimation, etc. [6, 7]. In recent years, there have been some methods to analyze and forecast financial data combined with computer technology. Among them, the financial forecasting technology based on deep learning has developed rapidly, such as the financial forecasting method based on the coupling of genetic algorithm and neural network and generalized regression neural network [8, 9]. With the continuous research of deep learning algorithm, convolutional neural network, deep trust network, and stack self-coding network are widely used in computer vision and

other aspects of modeling and get very good prediction results. In addition, there are many applications of traditional prediction methods in computer technology, such as SVM method and EMD method. In the financial analysis and prediction methods combined with computer, both the traditional prediction methods and the deep learning financial prediction methods based on convolution neural network have strong innovation and have high accuracy and reliability in specific fields or environments [10].

Based on the above analysis scheme of financial data combined with economics, mathematics, and computer science, even the prediction method is based on convolution neural network (CNN) or traditional artificial intelligence. If the model is built on the historical data of long time series, the memory and analysis ability of the model on the time dependence of data is poor, which cannot meet the requirements of financial data analysis on long time series. This system creatively uses the long-term and short-term memory neural (LSTM) network of the recurrent neural network (RNN) to build the financial prediction model. Its characteristics are as follows: the prediction model is to fit the change characteristics of historical data produced by mutual influence on the time line, rather than the data itself; that is, the data change of the day is related to yesterday's data, yesterday's data is related to the day before yesterday's data, and the day before yesterday's data is related to more distant data. The recursive relationship is analyzed by this kind of deduction, and the data series tracked is longer. This method makes full use of the characteristics of time series, relevance, and recurrence of financial data changes. Through big data, machine learning, and deep learning, we can integrate data, information, and decision-making in the market, realize intelligent association between data, assist users to extract key features between data, avoid market risks, and improve work efficiency and investment ability. The data analysis platform for quantitative trading designed and implemented in this paper has carried out demand analysis, design, implementation, and test. From the perspective of function test and performance test, two functions of similar stock search and financial matching model are selected and tested, and the results are in line with the expected results.

2. Design of Data Analysis Platform for Quantitative Transaction

The data analysis platform for quantitative transaction has many functional requirements, mainly divided into front-end page and back-end function. On the one hand, it provides an interactive platform to receive user requests and submit them to the background; on the other hand, it divides the background implementation into multiple functional submodules based on the design principle of low coupling, so as to ensure the clear division of labor and mutual cooperation between modules and realize the functions of similarity search and model matching [11, 12].

2.1. Overview of System Architecture. According to the above definition of functional requirements, the system is divided into four modules: user interaction module, data cleaning,

calculation engine, and data acquisition engine. The user interaction module is responsible for displaying relevant information to users, obtaining relevant parameters from users, intercepting relevant subsequences, and displaying the calculation results [13]. Data cleaning processes and normalizes the outliers in the API returned data, takes the original time series intercepted by customers as the input, and returns the results of similar stock retrieval and financial model matching to the user interaction module after calculation. The data acquisition engine is responsible for acquiring the historical price data of the past three years in the initialization phase, and then updating the database every day according to the trading situation of the day, so as to make sufficient data reserves for the calculation engine.

As can be seen from Figure 1, when the user contacts the system for the first time, the user first selects the sequence through the user interaction system and calls the interface in the data acquisition module to obtain the data of the subsequence required by the user, returns it to the user interaction system, and presents it to the user for the user to intercept the subsequence. When the user determines that the content to be searched is similar stock recognition or financial model matching, the user interaction system calls the corresponding interface in the computing engine according to the user's choice and presents the results to the user.

The function of user interaction module includes three parts: accepting user search request, calling other interfaces, and displaying results. Among them, receiving user search request includes receiving user stock sequence query request and allowing users to intercept subsequence and define query content. Calling other interfaces includes calling wind interface to judge the validity of stock symbol, calling wind interface to get related stock sequence, calling data cleaning interface, and calling two interfaces in calculation engine to get calculation results. The result display includes showing the query results of stock series to users and showing the calculation results of calculation engine to users.

2.2. Data Processing Module. The data acquisition engine module is mainly used to request the stock data from the third-party data acquisition interface and store it in the database to provide real-time, comprehensive, and accurate data guarantee for the calculation engine [14]. It includes query and acquisition of user specific data, one-time historical data acquisition in the initialization stage, daily data update and recalculation of dimension reduction results in the future, including APCA dimension reduction results, APCA dimension reduction results, and extreme point dimension reduction results, and finally updating the time series feature columns in the database according to the latest dimension reduction results. The main functions are as follows:

- (1) At the time of initialization, the daily data of 300 stocks are captured and stored in the database.
- (2) After initialization, the API is called daily to add the latest price time information of 300 stocks, and the

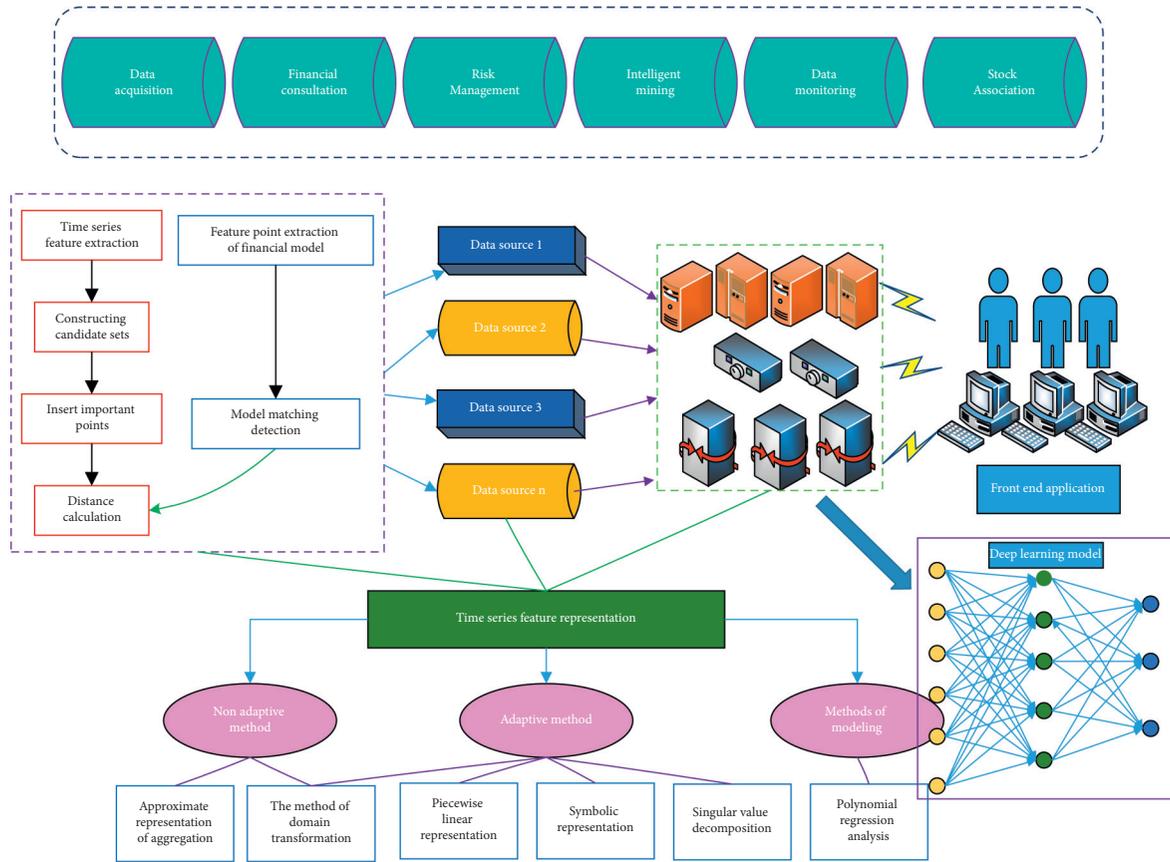


FIGURE 1: System architecture diagram.

interface of data cleaning module is called to check the integrity of the returned data and clean the data.

- (3) According to the needs of model matching method, query specific financial technical index data is obtained.
- (4) After the latest APCA dimensionality reduction results, the results are obtained and written into the database.
- (5) Refresh after the latest dimension reduction result of extreme point.
- (6) The feature columns are updated according to the results of extreme points.

Community is mainly for data preprocessing, first to determine whether the time series data is qualified, whether there is a phenomenon such as suspension; secondly to remove singular points and special points, modify the wrong data, fill in invalid data, and finally normalize it, as shown in Figure 2.

Data cleaning accepts the data from the wind data acquisition API sent by the user interaction module to clean and judge it. First, judge whether the stock has been suspended. If the highest price is equal to the lowest price, it proves that the stock is suspended and returns the user's relevant error information. If there is no suspension, judge whether there is an outlier point, such as 0 being negative, or the stock with a one-day increase of more than 10%, or the

stock with a decrease of more than -10% . In case of abnormal value, sina tushare interface should be called to conduct a second query to get its true value. If both queries are abnormal values, an error will be reported to the user. If the data pass the test, the stock sequence is normalized by min max to remove the difference caused by the different value of different stocks. The final result is the real value without outliers and missing data, including the external interface.

2.3. Hybrid Financial Identification Model

2.3.1. Financial Feature Point Filtering. Similar stock recognition model based on dynamic regularization distance mainly includes feature point extraction of financial model and model matching degree detection. As mentioned above, the financial model can only judge the relationship between the technical indicators of its local minima, so the input needs to be the local extremum required by the financial model. Any residual nonspecific point caused by small fluctuation or insufficient dimensionality reduction will affect the recognition of the time series if it enters the model judgment. The model will compare the noise points as feature points, which will affect the model recognition of several time series feature points, so we filter the feature points on the basis of APCA and extreme point dimension reduction [15]. Filter out the small peaks and troughs that

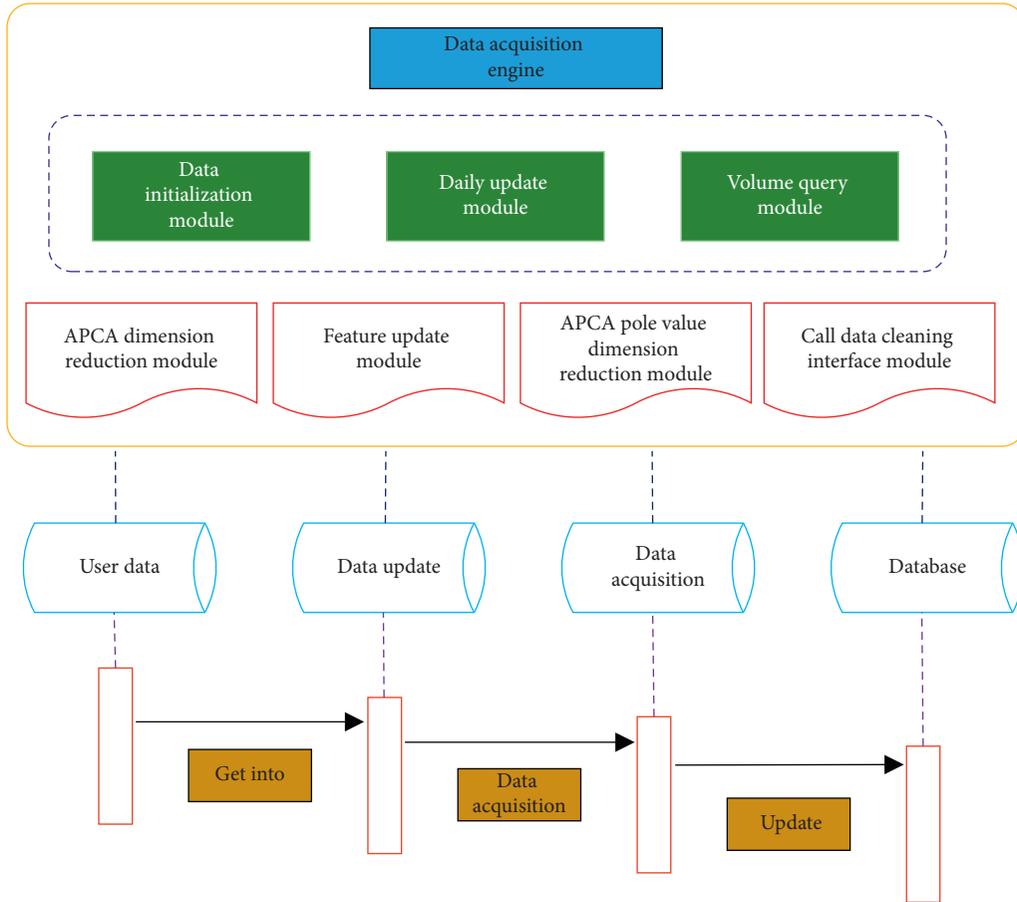


FIGURE 2: Schematic diagram of data acquisition engine function module.

affect the judgment of financial model, and extract the feature points that can be used to judge the financial model. Because the time requirement of the financial model is basically more than one month and three months, and the gap between the adjacent extreme points should be large, so if the two adjacent extreme points are close and the difference is small, we should ignore this pair of extreme points. The result is to meet the user-defined feature points of financial model, which will be input into the matching method of financial model.

2.3.2. Model Building. Financial models are often composed of several characteristic points, such as the local extremum in a certain period of time, and the relationship between its adjacent extremum points, including the relationship between values, time, and other technical indicators. And the financial model is often strict, so the traditional DTW comparison method is not reliable. Therefore, we need to model several common financial models and transform the description of feature points in the financial model into a programmable method, so as to traverse and match the time series and find the subsequence that meets the requirements of the model. Three most common classical K-line shapes, double top, head shoulder top, and double bottom, are used in the matching method. These three models appear

frequently in the stock price time series and are the most classic models. The input of the matching method is several local minimax feature points of the time series, and the output is the result of whether the time series is consistent or not.

3. Deep Learning Algorithm for Financial Data Analysis

3.1. Structure and Construction Method of Depth Model. Because neural network can simulate human brain to study complex problems, RNN, as a kind of recurrent neural network algorithm, has been widely used in time series analysis in recent years. On the basis of the traditional neural network, the neurons of recurrent neural network are connected between layers and nodes to ensure that the output information of the upper layer is the input information of the next layer. The specific performance is that the recurrent neural network will remember the previous information and apply it to the current output calculation and use the memory function characteristics of the model to affect the changes of recent data. Where $(x_1, x_2, x_3, \dots, x_n)$ is the input variable, $(w_1, w_2, w_3, \dots, w_n)$ is the weight, bias is the bias, Σ is the combination function, and $f(x)$ is the nonlinear activation function:

$$\begin{aligned} f(x) &= \sum w_i x_i + \varphi(t), \\ g &= g(f(x)). \end{aligned} \quad (1)$$

Single layer perceptron has limited function and even cannot solve the simple XOR operation, so based on the single layer network, a recurrent neural network, namely, feedforward neural network, is proposed. However, it is found that the output of RNN depends not only on the input information of the current network model, but also on the state of the previous network model. When the input time series data is too long, the derivation chain is too long, RNN network is difficult to preserve the characteristics of time series data, and it is prone to gradient disappearance or gradient explosion. The common RNN network structure cannot solve the problem of data dependence for a long time. Financial data is a kind of time series. With the further rapid development of the financial market and the accumulation of massive data, there is a long-term dependence between financial data. Therefore, a better neural network is needed to solve the problem of long-term dependence.

Long short-term memory model (LSTM) is a new network structure based on RNN, which has excellent performance in the related fields of time series analysis. It can solve the problem of RNN model gradient vanishing and accurately model the short-term or long-term data. At the same time, it is also the expansion component of LSTM which has gradually developed into a common deep neural network. There is no essential difference between the topology of LSTM network and RNN network. A more refined internal processing unit is used to store and update the context data transferred between hidden layers, and a gate control unit is used to control the flow of data information. LSTM network model includes input layer, output layer, and hidden layer. As a special memory module of LSTM model, the gate control unit of hidden layer stores the input and output information of all nodes and remembers the state information of current hidden layer. LSTM belongs to time series data (time-dependent) and requires global processing; the correspondence between input and output elements (such as between words) may have a large time span; the data is not too short but not too long, i.e., less than 1000 steps.

No matter RNN network model or LSTM, they all have the limitation of processing fixed length sequence data. In order to solve the problem of indefinite length sequence data, this paper proposes a Seq2Seq model, which maps one sequence as input to another output sequence and mainly completes two basic steps: encoding input and decoding output. The encoder encodes according to the input sequence data and outputs the encoded state at the end of the network. The decoder receives the last state of the encoder and takes it as the initial state. At the same time, the output value of each decoding step is the input value of the next step. Therefore, the training process of Seq2Seq model can better learn the temporal relationship between data and also can maintain stronger robustness. After all kinds of input sequences x with different lengths are mapped to the high-dimensional feature space, all the information in the

input layer of the whole sequence is compressed into a fixed length feature vector C through the encoder layer transmission. Vector C is usually the last hidden node h in RNN, or the weighted sum of multiple hidden nodes. The formula is as follows:

$$\begin{aligned} h(s) &= g(x_s, h_{s-1}) + \Delta\phi, \\ c &= \phi(\{h_1, h_2, h_3, \dots, h_t\}). \end{aligned} \quad (2)$$

The decoding layer uses the feature vector of the code and generates the output sequence of the tag. This layer is usually built with RNN architecture.

$$\begin{aligned} P_t &= f(y_{t-1}, S_{t-1}, c) + f(x_s, h_{s-1}) + \Delta\phi, \\ p(y_t | y < t, X|Y) + \Delta s &= f(y_{t-1}, S_t, c). \end{aligned} \quad (3)$$

When the fixed sequence is used, the codec framework is used to train the whole model to maximize the probability of the target sequence. However, due to the limitation of the amount of information contained in the feature vector of the fixed length coding, the performance of the Seq2Seq model will decrease with the increase of the length of the source sequence. To solve this problem, in the attention mechanism, the decoder tries to find the position of the most important information in the encoder sequence and uses the information and the previously decoded word to predict the next tag in the sequence.

The source sequence $x = (x_1, x_2, x_3, \dots, x_n)$ is input into the model in the forward and reverse directions, respectively, and then the positive and negative hidden nodes are obtained. The eigenvector S is obtained by weighting the hidden node h in RNN with different weights a :

$$S_i = \sum_{m=1}^{N_i} c_{ij} g_m + \ln e^{-m}. \quad (4)$$

Each hidden node h contains the corresponding input character X_i and its relation to the context. The higher the value of c_{ij} is, the more attention the i -th output has on the j -th input. According to this memory characteristic, the model can break through the limitation of fixed length sequence, construct different number of hidden nodes through different length sequence, and adjust the output of network model.

3.2. Prediction Model Design Based on Deep Learning. The time is used as the index to arrange, and the individual stocks and indexes are selected to contain multiple groups of trading data including opening price, closing price, the highest price, the lowest price, and trading volume. Generally, the closing price is an important factor representing the change of stock trend, so in the deep learning prediction model, the closing price is taken as the key feature, and the time is taken as the label to establish the deep learning model, and the closing price and the corresponding trend chart are predicted and output. In this paper, we first design and optimize the structure of LSTM and then compare LSTM with other neural networks.

3.2.1. Network Structure Design. Time series prediction is based on the characteristics of a period of time in the past to predict the data characteristics in the future. With the increase of the length of time series data, it is prone to the phenomenon of gradient disappearance. The long and short time memory model contains a special gate control part, so it has certain memory characteristics for the characteristics between the data. Therefore, according to the characteristics of time series data, this experiment builds LSTM network. The specific structure diagram is shown in Figure 3.

As shown in the figure above, the LSTM network adds a door control unit between the hidden layers. The gate control mechanism can control the transmission of information between neural networks. When the information passes through LSTM neural network, the gate control mechanism determines how much information needs to be retained or discarded to the memory unit, and the information in the new state needs to be saved to the memory unit. This makes the input and output of LSTM neural network have a long time series relationship.

3.2.2. Selection of Network Activation Function. In order to enhance the nonlinear characteristics of the neural network, the traditional LSTM network selects \tanh function as the activation function. With the gradual increase or decrease of the independent variable, the value of the corresponding function will also keep monotonic increase or decrease in the range of -1 and 1 . However, when the amount of training data is large, the gradient disappears easily, resulting in the network parameters being unable to be updated and the training of deep network unable to be effectively completed.

In order to avoid the problem of gradient vanishing, this experiment selects ReLU function as the activation function of LSTM network model, because the main function of this function is to improve the sparse characteristics of the network. In the part of ReLU function where the independent variable is less than 0 , all the function values become constant zero, which is called unilateral suppression. When the independent variable is greater than 0 , the value of the function increases with the increase of the independent variable. Compared with other activation functions, because of the unilateral suppression of ReLU, the neural network is sparser, which reduces the interdependence between data and alleviates the overfitting phenomenon. Figure 4 below is the schematic diagram of ReLU and \tanh functions.

According to the comparison between ReLU function and \tanh function, the value of function corresponding to ReLU increases with the increasing of independent variable. Although the corresponding value of \tanh function increases, with the increase of independent variables, its function value gradually approaches 1 . Compared with \tanh function, the convergence speed of ReLU function is faster. Therefore, after realizing neural network sparseness through ReLU, it can better mine the relevant characteristics of data, so that LSTM network can better fit the training data.

3.2.3. Network Training Method and Optimization. Gradient descent is an important method to adjust the parameters of neural network model. Its main goal is to minimize the value of loss function. But the gradient descent can only reach the local minimum. When the more complex network model has large amount of computation and long training time, the gradient descent has poor convergence for the model. Stochastic gradient descent algorithm, in order to speed up the iterative speed, selects the fastest direction to reduce the loss function for parameter updating and iterative calculation, until the neural network loss function value drops to meet the requirements of the range, but the stochastic gradient descent algorithm cannot achieve global optimization.

Compared with the above two gradient descent algorithms, batch gradient descent algorithm can reduce the number of updating parameters and training time. The algorithm divides the training data into small batch samples and updates the loss function of the training data in the current batch each time. Generally, the training batch is called batch, as shown in Figure 5 below.

Among them, the value table of n is not batch size, which is usually used to describe the size of batch. When the n value of the selected batch is small, it is difficult to achieve convergence. Similarly, when the selected n value is large, each training of the model takes a lot of time. Therefore, considering the training time and convergence of the model, it is necessary to set a reasonable range of n value.

When the training accelerates the model learning, the learning rate further determines the gradient moving direction of the batch gradient descent method. Using Adam algorithm to build the learning rate of LSTM network model is essentially to calculate the first-order moment estimation and second-order moment estimation of gradient and adaptively adjust the learning rate of different parameters. Therefore, in the iterative calculation of neural network, Adam can achieve model convergence at a faster speed according to the set range of learning rate. The formula of Adam optimizer is as follows.

4. Experiment and Analysis

In this paper, different super parameters in the algorithm are adjusted, and a number of experiments are carried out to improve the algorithm. RNN (recurrent neural network) prediction results will be closer to the target results with the increase of training times, but with the increase of training times, the prediction performance will not improve to a certain extent and tend to saturation. As shown in Figure 6, the prediction results under different training times are given.

In Figure 6, the blue line represents the original price, the yellow line represents the training forecast price, and the green line represents the simulation test forecast price. Among them, when the number of trainings is one, the prediction result data is seriously separated from the target data. When the number of trainings is 100, the prediction result is closer to the target result, and the prediction effect is significantly improved compared with one training. When the number of trainings is 1000, the prediction effect is not

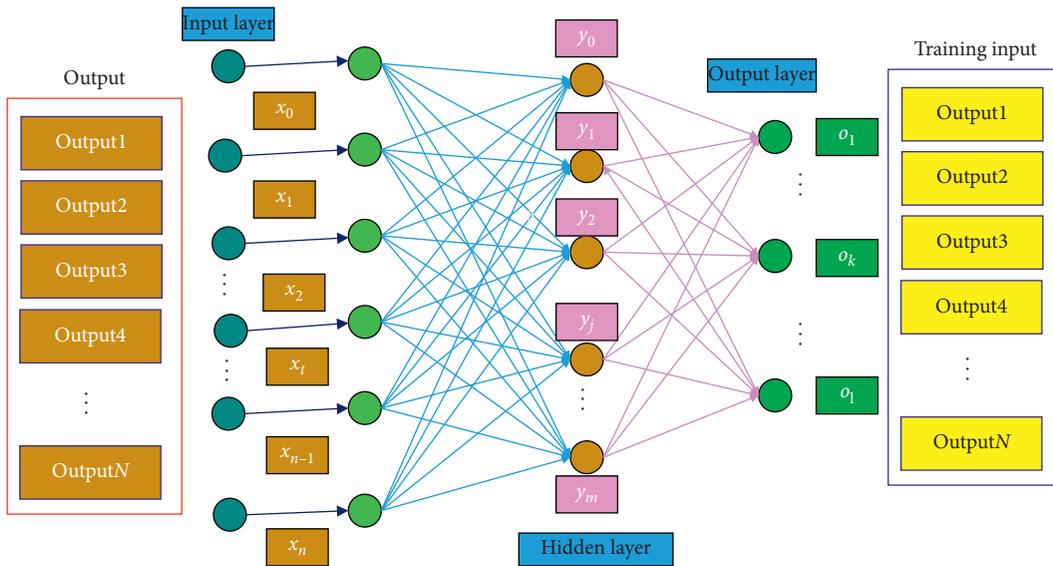


FIGURE 3: LSTM network structure diagram.

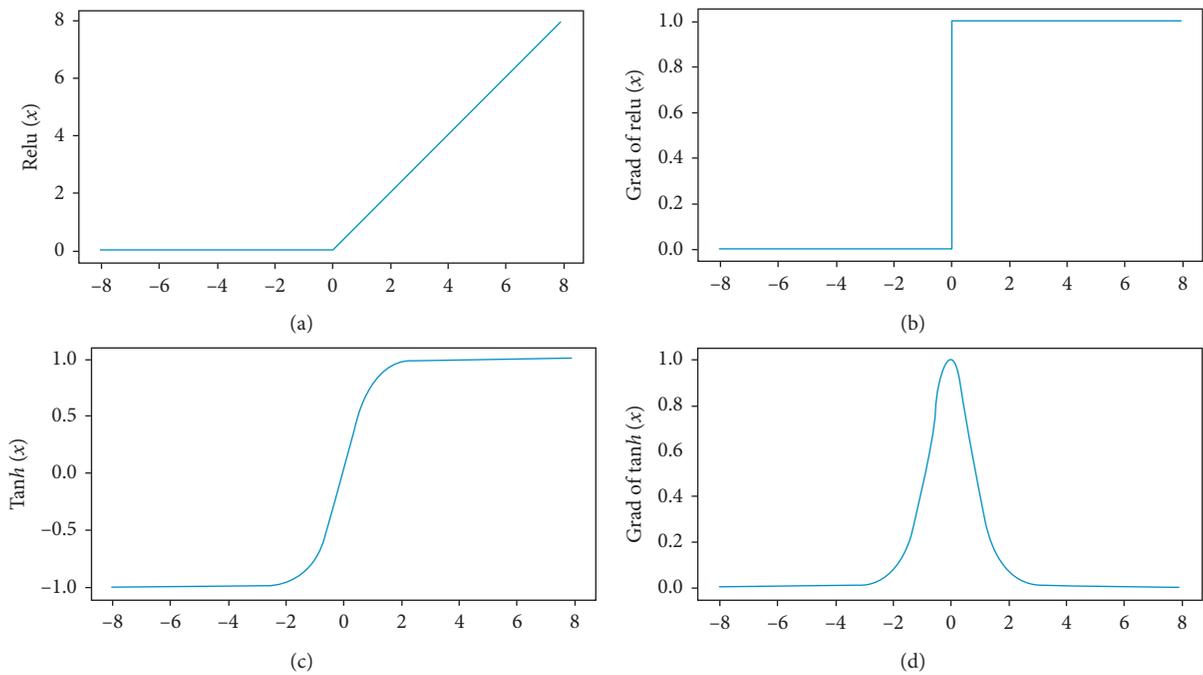


FIGURE 4: Comparison between ReLU function and tanh function.

significantly improved compared with 100 trainings, indicating that the prediction effect has been saturated and there is no need to increase the amount of training.

This paper selects the opening price, closing price, the highest price, the lowest price, and trading volume series of 10 stocks and indexes from 2012 to 2019 as the data set of the model. Using 70% of the data as the training set and 30% of the data as the verification set, the neural network model is evaluated. Finally, RMSE and Mae are selected as the evaluation indexes to evaluate the short-term closing price series. Table 1 and Figure 7 show the experimental prediction results.

Through the experiment, we can see that the prediction error of most samples is at a low level, which means that the model can get more realistic prediction results. However, when there is a large fluctuation between the data, the prediction model will also be inaccurate, resulting in large prediction error. Shanghai Pudong Development Bank is a special component stock data among 10 groups of data. Its closing price ranges from several hundred to tens of thousands. When the long and short memory model (LSTM) predicts the sample, the model is inaccurate. For the second group of experiments, RNN, LSTM, and Seq2Seq neural networks are selected to analyze the short-term prices of 50

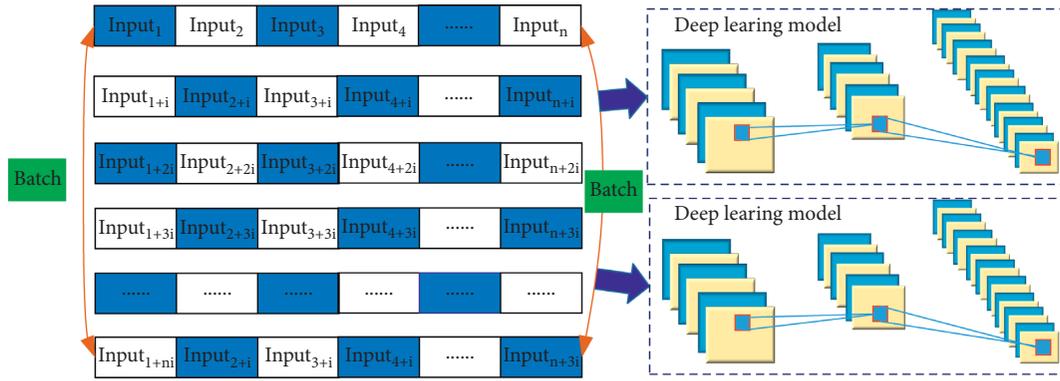


FIGURE 5: Model training input after batch division.

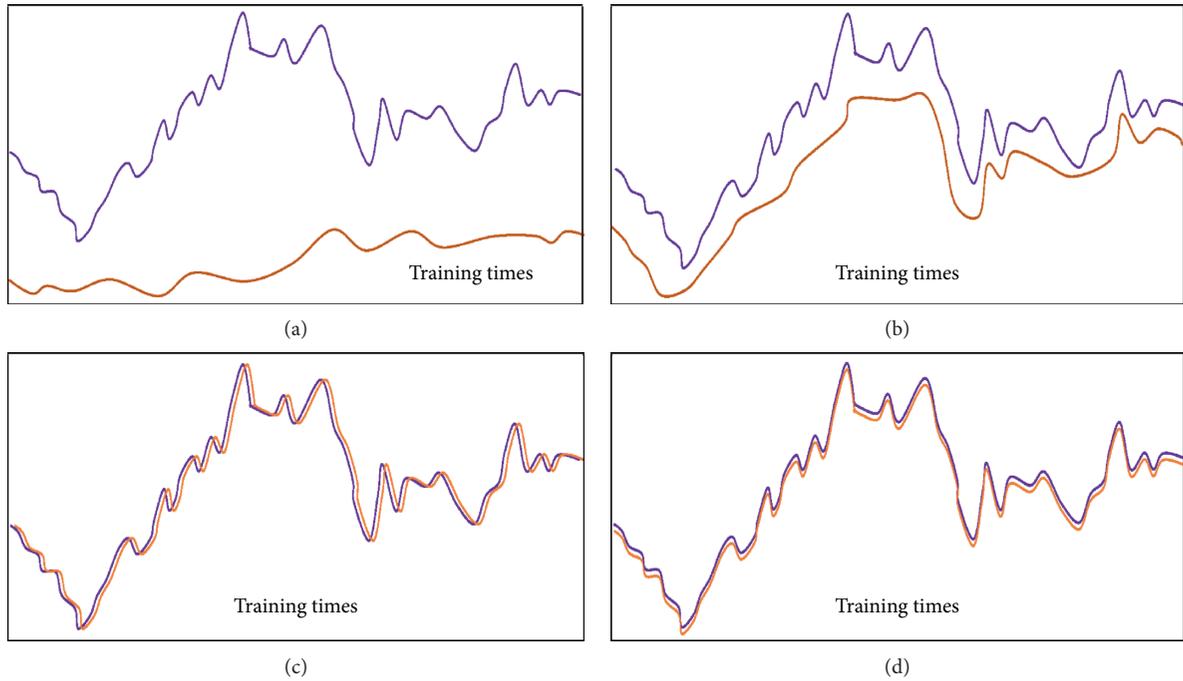


FIGURE 6: Training frequency and effect.

TABLE 1: LSTM network prediction results.

| Experimental sample | Sample type | True value | Estimate value | RMSE | MAPE |
|---------------------|-------------------|------------|----------------|--------|--------|
| 102005 | Index | 23.32 | 23.42 | 0.0096 | 0.0215 |
| 398123 | Index | 12589.5 | 12645.8 | 0.0005 | 0.0291 |
| 601122 | Index | 22.39 | 21.89 | 0.0719 | 0.0273 |
| 601127 | Index | 17.25 | 17.29 | 0.0005 | 0.0286 |
| 601142 | Individual stocks | 31.38 | 31.94 | 0.0002 | 0.0223 |
| 601147 | Individual stocks | 43 | 43.52 | 0.0004 | 0.0227 |
| 601620 | Individual stocks | 920 | 920.28 | 0.0002 | 0.0212 |
| 602277 | Individual stocks | 29.6 | 28.6 | 0.0006 | 0.0283 |
| 602438 | Individual stocks | 91.93 | 81.53 | 0.0002 | 0.0232 |
| 602439 | Individual stocks | 16.88 | 17.21 | 0.0003 | 0.0224 |

stocks in Shanghai Stock Exchange, in order to show the performance of each algorithm more clearly. A single stock of Huaxia happiness is selected to show the prediction results.

According to the experimental results, we can see that the deep learning algorithm model can track the price trend of learning stock data, but with the growth of time, all kinds of network models have some fluctuations in the price

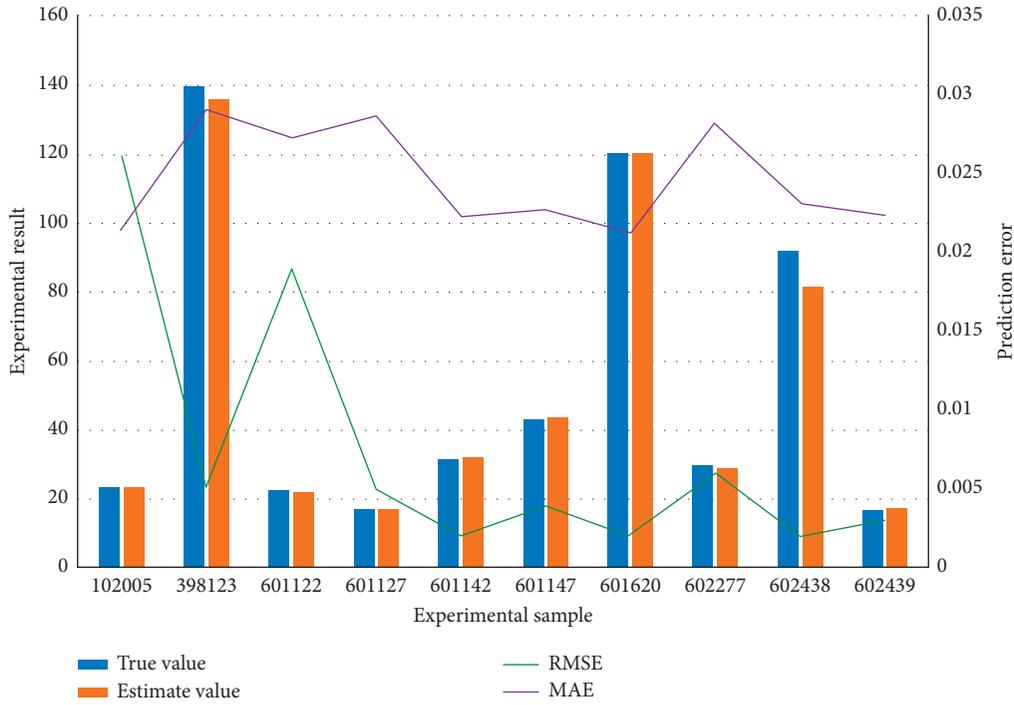


FIGURE 7: Network prediction results.

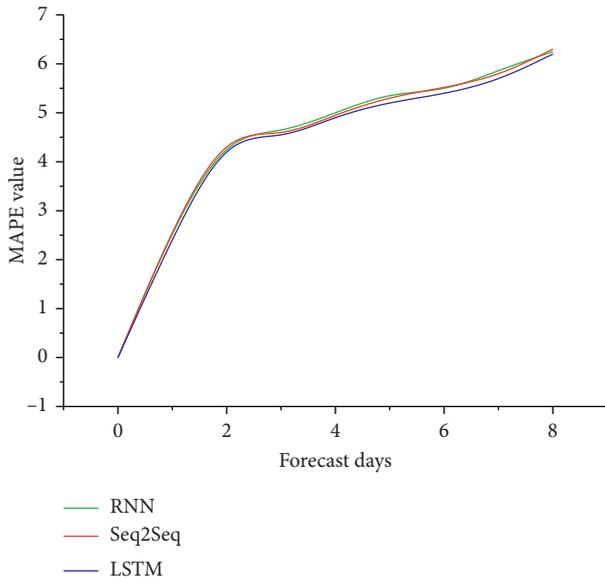


FIGURE 8: MAPE evaluation index.

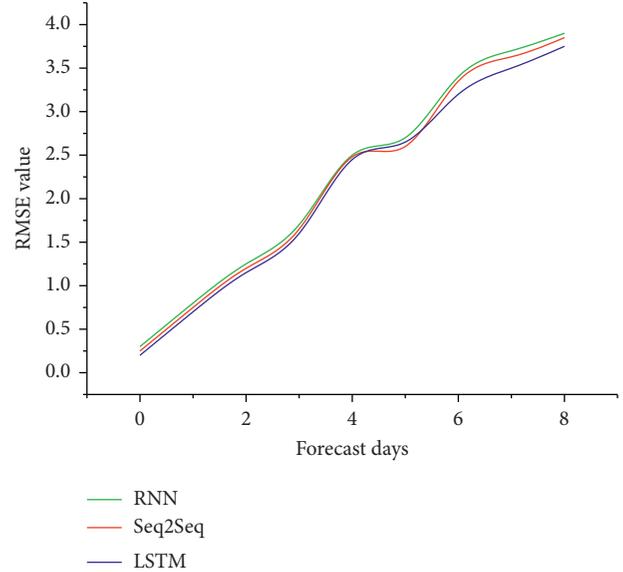


FIGURE 9: RMSE evaluation index.

prediction. The red part is LSTM model, which can predict the stock price deviation slightly lower than Seq2Seq and RNN. RMSE, MAPE, and Mae are selected to evaluate the prediction effect of three network models, as shown in Figures 8 and 9.

With the increase of forecasting days, the forecasting error of the three deep learning models is also gradually increasing. Compared with other Seq2Seq and RNN networks, the LSTM network model is relatively small. Therefore, several stocks are selected for comparative

analysis, and the forecast results are shown in Table 2 and Figure 10.

Through the second group of experiments, when the number of neural network cycles is fixed, with the increase of the number of time series data samples, RMSE and MAPE indexes are increased in varying degrees, which also shows that the accuracy of LSTM model in short-term stock prediction is gradually reduced, and the prediction error is increasing. LSTM network with lower complexity is better than Seq2Seq model with higher complexity.

TABLE 2: Prediction results of three network models.

| Experimental sample | RNN | | LSTM | | Seq2Seq | |
|---------------------|--------|--------|--------|--------|---------|--------|
| | RMSE | MAPE | RMSE | MAPE | RMSE | MAPE |
| 001 | 0.0586 | 0.0131 | 0.0562 | 0.0131 | 0.0571 | 0.0122 |
| 002 | 0.0647 | 0.0132 | 0.0494 | 0.0133 | 0.0593 | 0.0134 |
| 003 | 0.0678 | 0.0143 | 0.0634 | 0.0138 | 0.0687 | 0.0144 |
| 004 | 0.0663 | 0.0141 | 0.0326 | 0.0115 | 0.0361 | 0.0117 |
| 005 | 0.0382 | 0.0118 | 0.0381 | 0.0118 | 0.0462 | 0.0123 |
| 006 | 0.0635 | 0.0138 | 0.0249 | 0.0112 | 0.0275 | 0.0113 |
| 007 | 0.0786 | 0.0156 | 0.0688 | 0.0144 | 0.0744 | 0.0151 |
| 008 | 0.0567 | 0.0131 | 0.0478 | 0.0124 | 0.0516 | 0.0127 |
| 009 | 0.0783 | 0.0156 | 0.0688 | 0.0144 | 0.0744 | 0.0151 |
| 010 | 0.0617 | 0.0136 | 0.0548 | 0.0228 | 0.0941 | 0.0179 |
| 011 | 0.0483 | 0.0126 | 0.0514 | 0.0127 | 0.0568 | 0.0131 |
| 012 | 0.0696 | 0.0145 | 0.0711 | 0.0147 | 0.0688 | 0.0146 |
| 013 | 0.0447 | 0.0122 | 0.0448 | 0.0122 | 0.0814 | 0.0158 |
| 014 | 0.0436 | 0.0121 | 0.0399 | 0.0119 | 0.0455 | 0.0122 |
| 015 | 0.0853 | 0.0166 | 0.0442 | 0.0122 | 0.0556 | 0.0121 |

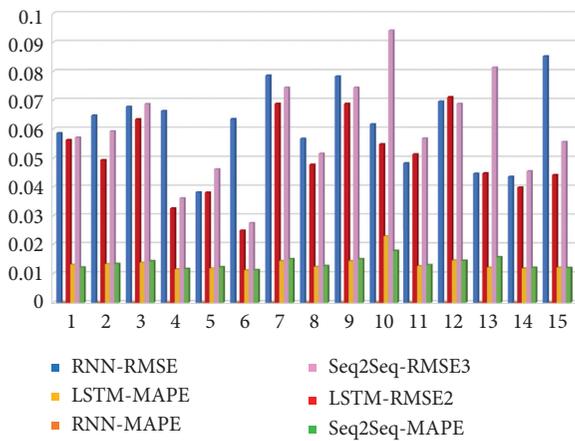


FIGURE 10: Comparison of prediction results of network models.

5. Conclusion

The data analysis platform for quantitative trading designed and implemented in this paper has carried out demand analysis, design, implementation, and test. From the perspective of function test and performance test, two functions of similar stock search and financial matching model are selected and tested, and the results are in line with the expected results. Based on the deep learning financial data analysis model, this paper introduces the characteristics and basic principles of the mainstream deep learning RNN, LSTM, and Seq2Seq algorithms. On the basis of these three algorithms, the prediction values of the three algorithms are compared to verify the advantages of deep learning algorithm in financial data analysis, and the deep neural network has higher accuracy in time series data prediction. This paper introduces the implementation details of the key modules of the platform, including web services, data cleaning, computing engine, and data acquisition engine. The user interaction module obtains and displays the retrieval results through data analysis, calling the background, and computing engine; the data cleaning module fills, optimizes, and

normalizes the data through business experience. The calculation engine module uses the algorithm and extracts the database information to get the similar time series and matching financial model; finally, the data acquisition module fills the database with historical data at the initialization stage and updates the database every day. The data analysis platform for quantitative trading designed and implemented in this paper has carried out demand analysis, design, implementation, and test. From the perspective of function test and performance test, two functions of similar stock search and financial matching model are selected and tested, and the results are in line with the expected results. Based on the deep learning financial data analysis model, this paper introduces the characteristics and basic principles of the mainstream deep learning RNN, LSTM, and Seq2Seq algorithms. On the basis of these three algorithms, the prediction values of the three algorithms are compared to verify the advantages of deep learning algorithm in financial data analysis, and the deep neural network has higher accuracy in time series data prediction.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no known conflicts of interest or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] F. Beck, S. Koch, and D. Weiskopf, "Visual analysis and dissemination of scientific literature collections with SurVis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 180–189, 2016.
- [2] W. Jiang, F. Lin, J. Zhang, T. Zhan, P. Cao, and S. Wang, "Deep-learning-based segmentation and localization of white matter hyperintensities on magnetic resonance images,"

- Interdisciplinary Sciences: Computational Life Sciences*, vol. 12, no. 4, pp. 438–446, 2020.
- [3] L. Zhao, F. Li, Y. Zhang, X. Xu, H. Xiao, and Y. Feng, “A deep-learning-based 3D defect quantitative inspection system in CC products surface,” *Sensors*, vol. 20, no. 4, pp. 980–996, 2020.
- [4] Y. Pan, Z. Shao, T. Cheng et al., “Analysis of urban waterlogging influence based on deep learning model,” *Wuhan Daxue Xuebao (Xinxi Kexue Ban)/Geomatics and Information Science of Wuhan University*, vol. 44, no. 1, pp. 132–138, 2019.
- [5] T. Muling, T. Muqin, Y. Jieming, and J. Li, “Optimization of RBFneural network used in state recognition of coal flotation,” *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 2, pp. 1193–1204, 2018.
- [6] Y. Chen, A. Janowczyk, and A. Madabhushi, “Quantitative assessment of the effects of compression on deep learning in digital pathology image analysis,” *JCO Clinical Cancer Informatics*, vol. 4, no. 4, pp. 221–233, 2020.
- [7] C. Atkinson-Clement, J. Sadat, and S. Pinto, “Behavioral treatments for speech in Parkinson’s disease: meta-analyses and review of the literature,” *Neurodegenerative Disease Management*, vol. 5, no. 3, pp. 233–248, 2015.
- [8] M. J. Albers, “Quantitative data analysis-in the graduate curriculum,” *Journal of Technical Writing and Communication*, vol. 47, no. 2, pp. 215–233, 2017.
- [9] M. Byra, A. Han, A. S. Boehringer et al., “Quantitative liver fat fraction measurement by multi-view sonography using deep learning and attention maps,” *The Journal of the Acoustical Society of America*, vol. 146, no. 4, p. 2864, 2019.
- [10] J. F. Zheng, J. D. Jiao, and L. P. Sun, “A modeling approach for early-warning of water bloom risk in urban lake based on neural network,” *Zhongguo Huanjing Kexue/china Environmental Science*, vol. 37, no. 5, pp. 1872–1878, 2017.
- [11] X. Du, R. Tang, S. Yin, Y. Zhang, and S. Li, “Direct segmentation-based full quantification for left ventricle via deep multi-task regression learning network,” *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 3, pp. 942–948, 2019.
- [12] Y. An and C. Shen, “Design and implementation of internet financial transaction risk identification system based on big data analysis,” *C e Ca*, vol. 42, no. 2, pp. 767–771, 2017.
- [13] M. Sun, I. Konstantelos, and G. Strbac, “A deep learning-based feature extraction framework for system security assessment,” *IEEE Transactions on Smart Grid*, vol. 10, no. 5, pp. 5007–5020, 2018.
- [14] R. Yan, L. Wang, Z. Hu et al., “Quantitative verification on satellite observational data of ionospheric structure parameters using ground-based data,” *Acta Seismologica Sinica*, vol. 39, no. 4, pp. 549–557, 2017.
- [15] P. Zhaodong, T. Ping, L. Liangkun et al., “Hierarchical decentralized control of building structure based on adaptive RBF neural network algorithm,” *China Civil Engineering Journal*, vol. 51, no. 1, pp. 51–57, 2018.